Listing of Claims:

1-25. (Canceled)

26 (Previously Presented). A method comprising:

searching data stored in a computer readable media for a first initial search result using at least a first portion of a first key; and

if the first initial search result is a route index corresponding to the first key, then returning the route index; and

if the first initial search result is a subtree index for an iterative search, then performing an iterative search of the data stored in the computer readable media, the iterative search comprising: searching the data for an iterative search result using a subsequent key comprising the subtree index found in a preceding search and at least a next portion of the first key; and if the iterative search result is a route index corresponding to the first key, then returning the route index; and if the iterative search result is a subtree index for a next search, then performing the iterative search again.

27 (Previously Presented). The method of claim 26, further comprising:

searching the data for a second initial search result using at least a first portion of a second key, wherein searching the data for the second initial search result is at least partially performed in parallel with searching the data for the iterative search result.

28 (Previously Presented). The method of claim 27, wherein the first key, the second key, or both the first and second keys comprise at least one of either a 32 bit IPv4 address or a 128 bit IPv6 address.

29 (Previously Presented). The method of claim 27, wherein the first key, the second key, or both the first and second keys further comprise a prefix corresponding to a Virtual Private Network identifier.

30 (Previously Presented). The method of claim 26, wherein the data is stored in a lookup table.

31 (Previously Presented). The method of claim 30, wherein the subtree index comprises a pointer to at least one other entry stored in the lookup table.

32 (Previously Presented). An apparatus comprising:

a memory storing data;

a forwarding engine configured to search the data for a first initial search result using at least a first portion of a first key, wherein the forwarding engine is configured to return a route index if the first initial search result is a route index corresponding to the first key, and wherein the forwarding engine is configured to perform an iterative search if the first initial search result is a subtree index, wherein the iterative search comprises: searching the data for an iterative search result based on a subsequent key comprising the subtree index found in a preceding search and at least a next portion of the first key; and if the iterative search result is a route index corresponding to the first key, then returning the route index; and if the iterative search result is a subtree index, then performing the iterative search again.

33 (Previously Presented). The apparatus of claim 32, further comprising:

a controller configured to at least partially search in parallel the data for the iterative search result and a second initial search result using at least a first portion of a second key.

34 (Previously Presented). The apparatus of claim 33, wherein the first key, the second key, or both the first and second keys comprise at least one of either a 32 bit IPv4 address or a 128 bit IPv6 address.

35 (Previously Presented). The apparatus of claim 33, wherein the first key, the second key, or both the first and second keys further comprise a prefix corresponding to a Virtual Private Network identifier.

36 (Currently Amended). The apparatus of claim 33, wherein the data is stored in a lookup table in the memory.

37 (Previously Presented). The apparatus of claim 36, wherein the subtree index comprises a pointer to at least one other entry stored in the lookup table.

38 (Currently Amended). An apparatus comprising:

a memorymeans for storing data;

means for searching the data for a first initial search result using at least a first portion of a first key, wherein said means is configured to return a route index if the first initial search result is a route index corresponding to the first key, and wherein said means is configured to perform an iterative search if the first initial search result is a subtree index, wherein the iterative search comprises: searching the data for an iterative search result based on a subsequent key comprising the subtree index found in a preceding search and at least a next portion of the first key; and if the iterative search result is a route index corresponding to the first key, then returning the route index; and if the iterative search result is a subtree index, then performing the iterative search again.

39 (Previously Presented). The apparatus of claim 38, further comprising:

means for at least partially searching in parallel the data for the iterative search result

and a second initial search result using at least a first portion of a second key.

40 (Previously Presented). The apparatus of claim 39, wherein the first key, the

second key, or both the first and second keys comprise at least one of either a 32 bit IPv4

address or a 128 bit IPv6 address.

41 (Previously Presented). The apparatus of claim 39, wherein the first key, the

second key, or both the first and second keys further comprise a prefix corresponding to a

Virtual Private Network identifier.

42 (Canceled).

43 (Currently Amended). The apparatus of claim [[42]]38, wherein the subtree index

comprises a pointer to at least one other entry in the means for storing the data.

44 (Previously Presented). A method comprising:

searching data stored in a computer readable media for an iterative search result using

a subtree index found in a preceding search of the computer readable media and at least a next

portion of a first key; and

if the iterative search result is a route index corresponding to the first key, then

returning the route index; and

if the iterative search result is a subtree index for a next search, then performing said

searching data for an iterative search result again.

45 (Previously Presented). The method of claim 44, further comprising:

searching the data for a second initial search result using at least a first portion of a second key, wherein searching the data for the second initial search result is at least partially performed in parallel with searching the data for the iterative search result.

46 (Previously Presented). The method of claim 45, wherein the first key, the second key, or both the first and second keys comprise at least one of either a 32 bit IPv4 address or a 128 bit IPv6 address.

47 (Previously Presented). The method of claim 45, wherein the first key, the second key, or both the first and second keys further comprise a prefix corresponding to a Virtual Private Network identifier.

48 (Previously Presented). The method of claim 44, wherein the data is stored in a lookup table.

49 (Previously Presented). The method of claim 48, wherein the subtree index comprises a pointer to at least one other entry stored in the lookup table.

50 (Previously Presented). The method of claim 26, wherein the data is stored in a lookup table comprising a plurality of mappers, the lookup table is configured to return in a single search of the lookup table a route index corresponding to a mapper key, and the iterative search is performed if a length of the first key is greater than a length of the mapper key and the first initial search result is a subtree index for an iterative search.

51 (Previously Presented). The method of claim 50, wherein the length of the mapper key is 40 bits and the length of the first key is at least 64 bits.

52 (Previously Presented). The method of claim 26, wherein the data is stored in a lookup table comprising a plurality of mappers including a direct mapper and at least one indirect mapper, searching the data for the first initial search result comprises searching both the direct mapper and said at least one indirect mapper using the first portion of the first key, and performing the iterative search of the data comprises searching again said at least one indirect mapper using the subsequent key without searching again the direct mapper.

53 (Previously Presented). The method of claim 26, wherein the data is stored in a lookup table comprising a plurality of mappers including a direct mapper and at least one indirect mapper and said at least one indirect mapper is configured to store data corresponding to at least two mapper levels.

54 (Previously Presented). The method of claim 26, further comprising:

receiving a command indicating that the iterative search should be performed using the subtree index found in the preceding search and the at least next portion of the first key.

55 (Currently Amended). The apparatus of claim 32, ~~further comprising:~~

~~a lookup table in communication with the forwarding engine,~~ wherein the data is stored in a lookup table in the memory, the lookup table comprises a plurality of mappers including a direct mapper and at least one indirect mapper, said at least one indirect mapper is configured to store data corresponding to at least two mapper levels, and the lookup table includes a pointer selector configured to cause a search of an appropriate mapper level to be

performed in said at least one indirect mapper based on a command received from the

forwarding engine.

56 (Currently Amended). The apparatus of claim 32, ~~further comprising:~~

~~a lookup table in communication with the forwarding engine,~~ wherein the data is

stored in a lookup table in the memory, the lookup table comprises a plurality of mappers, the

lookup table is configured to return in a single search of the lookup table a route index

corresponding to a mapper key, and the iterative search is performed if a length of the first

key is greater than a length of the mapper key and the first initial search result is a subtree

index for an iterative search.